IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **LeRoy W. Tilt IV and Matthew B. Trevathan**	Confirmation No.: **8815**
Serial No.: **09/823,331**	Examiner: **James Lee Brown**
Filed: **March 30, 2001**	Group Art Unit: **2144**
For: **METHOD AND APPARATUS FOR SERVER SIDE QUEUING TO CONTROL PAGE PRESENTATION ON WEB ENABLED DEVICES**	Attorney Docket No. **RSW9-2001-0019-US1**

FILED ELECTRONICALLY

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Attention: Board of Patent Appeals and Interferences

CORRECTED APPELLANTS' BRIEF

Sir:

This brief is in furtherance of the Notice of Appeal filed in this case on or about April 25, 2005.

1) REQUIRED FEE

The requisite fee of $500.00 set forth in §41.20(b)(2) is submitted herewith. If the submitted fee is insufficient, the United States Patent and Trademark Office (hereinafter "Office") is authorized to charge Applicant's Deposit Account No. 19-5425 for any shortfall.

2) REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation. Accordingly, International Business Machines Co. is the real party in interest.

3) RELATED APPEALS AND INTERFERENCES

The appellant, assignee, and the legal representatives of both are unaware of any other appeal or interference that will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

4) STATUS OF CLAIMS

a. Claims canceled: 10

b. Claims withdrawn from consideration but not canceled: None

c. Claims pending: 1-9, 11-22

d. Claims allowed: None

e. Claims merely objected to: None

f. Claims rejected: 1-9 and 11-22

g. Claims appealed: 1-9 and 11-22

Appealed claims 1-9 and 11-22 as currently pending are attached as Appendix A hereto.

5) STATUS OF AMENDMENTS

There are no un-entered amendments to the specification claims or drawings in this case.

6) SUMMARY OF CLAIMED SUBJECT MATTER

The present invention, as best described and shown on pages 10, line 1 through page 11, line 10 and page 11, line 1 through page 12, line 7 and in Figure 4, is a method and apparatus for loading web pages, including supplemental files such as pictures, sound files, video files, etc., at a browser.

One of the problems of the prior art addressed by the present invention is that browsers typically read the HTML code in a Web page from left to right and from top to bottom. Accordingly, the browser encounters the embedded references to such supplemental files in the order in which they are encountered while reading the page. The browser will send requests back to the server for those supplemental files in the order that the browser encounters the references while reading the HTML code. Since a browser has a limited number of ports, the supplemental files may not be retrieved and loaded in the most efficient manner. For instance, if a browser has four ports and the requested page has 14 supplemental files, in which the first four referenced supplemental files are large files and the next 10 are small files, the browser may take a long time to download the first four large files, while the person sitting at the client

browser watches a largely or completely blank screen. If the ten small files could be downloaded first, the browsing experience for the person can be much improved because he/she could then have something to look at while waiting for the four large files to download.

The present invention addresses this concern without the need to modify the browser software in any way. In accordance with the invention, the order in which supplemental files referenced in a Web page are downloaded from the server to the requesting client is specified by the designer of the HTML code of the Web page and controlled at the server side regardless of the order in which the client-side Web browser encounters and requests the supplemental files. Specification, page 9, lines 17-21. Particularly, each supplemental file referenced in a Web page has a sequence number associated with it. In a preferred embodiment, the sequence number is provided as an additional attribute of the tag that calls the supplemental file. Specification, page 10, line 5 – page 11, line 5. Since the client Web browser is a standard Web browser, it will have no idea what the sequence attribute is, which is irrelevant because a browser will simply ignore any attribute within a tag that it does not understand. Specification, page 10, lines 20-22 and page 12, lines 3–7. However, at the server-side, when the page is requested, the server parses the page before sending it to the requesting client to find the tags for the supplemental files embedded within the page and reads the associated sequence number attributes. It then builds a queue for serving the supplemental files to the client machine, the supplemental files being

queued in the order dictated by the sequence numbers. Specification, page 10, line 22 – page 11, line 5

Thereafter, regardless of the order in which the browser returns requests for the supplemental files, the server will serve the supplemental files in the order dictated by the queue. Existing browsers already are equipped to receive and cache files and associate such cached files with files referenced in an HTML page. Accordingly, the fact that the supplemental files referenced in a Web page may be received in an order different from the order in which the browser requests them is of no consequence. Specification, page 6, lines 8–13. Accordingly, the invention resides entirely at the server side and will work with any Web browser.

Independent claims 1, 12, and 19 recite the overall inventive technique employed, for instance, at the server. All three claims are rather similar in scope. On the other hand, independent claim 9 claims a computer file, e.g., a Web page, constructed in accordance with the present invention. The table below maps the specification and drawings onto the independent claims.

Description in specification and drawings	**Claim 1**	**Claim 9**	**Claim 12**	**Claim 19**

Page 10, lines 1-11.	1. A method of serving a Web page to a requesting client, said Web page comprising code defining said Web page and including a plurality of supplemental files, said method comprising the steps of: first code at least partially defining said Web page, said code including a plurality of references to supplemental files containing content of said page; and	9. A computer readable storage medium containing executable code for controlling a computer for rendering a Web page, first code at least partially defining said Web page, said code including a plurality of references to supplemental files containing content of said page; and	12. A computer program product embodied on computer readable media readable by a computing device, said product for serving Web pages to a requesting client machine, wherein at least one of said Web pages contains a plurality of references to supplemental files comprising content of said Web page, *e*, said references including order data indicating an order in which said supplemental files are to be served relative to said other supplemental files contained in said page, said product comprising:	19. A system for serving Web pages to a requesting client machine, at least one of said Web pages containing a plurality of references to supplemental files comprising content of said Web page, said page including order data indicating an order in which said supplemental files are to be served relative to said other supplemental files contained in said page, the system comprising:

Page 10, lines 11-20. See also, page 10, line 22 through page 11, line 5.		second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file.		
Figure 4, step 401 and page 14, lines 4-7. See also, page 10, line 22 through page 11, line 5.			first computer readable program code for receiving requests for said Web pages;	a computer including memory, and a processor, the memory being accessible by the processor and storing computer-readable programming including, first computer readable program code for receiving requests for said Web pages;

Figure 4, step 403 and page 14, lines 7-10. See also, page 10, line 22 through page 11, line 5.			second computer readable program code for obtaining code defining said requested Web pages responsive to said requests;	second computer readable program code for obtaining code defining said requested Web pages;
Figure 4, step 405 and page 14, lines 10-16. See also, page 10, line 22 through page 11, line 5.	parsing said code defining said Web page to detect order data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page;		third computer readable program code for parsing said code defining said Web page to detect said order data;	third computer readable program code for parsing said code defining said Web page to detect said order data;

Figure 4, step 407 and page 14, lines 17-20. See also, page 10, line 22 through page 11, line 5.	constructing a queue indicating said order;		fourth computer readable program code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order;	fourth computer readable program code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order;
Figure 4, step 409 and page 14, lines 20-24. See also, page 10, line 22 through page 11, line 5.	serving said code to said requesting client;		fifth computer readable program code for serving said code defining said Web page to said requesting client machine;	fifth computer readable program code for serving said code defining said Web page to said requesting client machine;
Figure 4, step 415 and page 14, line 24 to page 15, line 7. See also, page 10, line 22 through page 11, line 5.	serving said supplemental files to said client in said order indicated in said queue.		sixth computer readable program code for serving said supplemental files to said requesting client machine in said order of said queue.	sixth computer readable program code for serving said supplemental files to said requesting client machine in said order of said queue.

7) GROUNDS FOR REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-9 and 11-22 stand rejected under 35 U.S.C. §103(a) as unpatentable over U.S. Patent No. 6,269,403 issued to Anders (hereinafter Anders) in view of U.S. Patent No. 6,073,124 issued to Krishnan et al. (hereinafter Krishnan).

8) ARGUMENT

A. The Anders Reference

Anders discloses a method and apparatus for serving Web pages, including supplemental files, to a requesting client. However, the method and apparatus disclosed in Anders is entirely different than that of the present invention. Unlike the present invention, Anders requires the Web browser software to be modified to function with the invention. See col. 8, lines 2-6 (which describes the need for a Jammer unpacker "on the client") and col. 8, lines 51-54, and col. 12, line 65 – col. 14, line 7 (which describes in detail the software needed at the browser to implement the invention).

Anders presumes to alleviate the problem of slow downloads resulting from the delay inherent in networks with slow upstream channels (the direction from the client browser to the server). Specifically, according to Anders column 3, lines 1-25, numerous HTTP transactions severely impact the data transfer performance of high-speed satellite and cable modem systems because, in these systems, the transmission speed from a server to a client, the "downstream" connection, is substantially higher than the transmission speed from client to server, the "upstream" connection. For a Web page having numerous objects, the slow speed "upstream" request connection limits data transfer performance due to the response latency of multiple HTTP requests for these objects.

Anders purports to address these performance limitations by improving the efficiency of object retrieval and transfer in multimedia computer environments. Specifically, in contrast to the multiple transactions currently required to retrieve multiple objects in HTTP, Anders reduces the number of transactions to retrieve a set of n objects from n to 1, thereby substantially reducing the latency due to slow start and cascaded round trip delays from opening multiple server connections. Anders includes data for multiple objects in a single data stream in an interleaved format. A "publisher" software module at the server stores and delivers object data interleaved with data definition entries identifying respective object data in a data format comprising a single stream. This data format eliminates the need for multiple, asynchronous transactions. Moreover, the publisher enables one to customize and optimize how the object data is prioritized and interleaved by defining a configuration template. A browser recognizes a data stream prepared in the data format of the present invention and unpacks the data stream customized by the publisher to achieve a desired effect on the viewer upon delivery of the multimedia data.

Anders' scheme is entirely different than Applicant's. With reference to Anders' Figure 8, the server transmits the requested page to the requesting client in a particular data stream format 190 that includes the data for the main object (the Web page) and the data for the supplemental objects (such as embedded pictures, etc.) in data entries (packets, such as packets 181-189) that are interleaved with each other in an order selected by the user. More particularly, the data stream 190 comprises a stream

header 180 at the beginning of the stream followed by data definition entries and HTML data entries. Each data definition entry, e.g., 181, 182, 185, 187, defines a supplemental object/file present in the Web page data stream. There is one data definition entry per object/file. The HTML data entries are the actual data of the objects/files (including the main file as well as the supplemental files). Each file will typically consist of many HTML data entries that the browser assembles together to render the whole file. The data definition entry that defines any given object/file must precede the first HTML data entry of that file in order for the browser to know what to do with those HTML data entries when it receives them. Col. 7, line 57 - col. 8, line 36.

The basic premise of Anders' invention is that the publisher 210 (Fig. 11) interleaves the data for the entire web page in a way dictated by user supplied data and serves it to the client that way. The browser, upon receiving each data definition entry, creates an entry in an unpacked object cache (UOC). Then, when the browser starts receiving the HTML data entries corresponding to the supplemental file identified by any given data definition entry, it will append that HTML data to the entry it created in its UOC. In Anders, the browser receives the tags identifying the supplemental files in the order dictated by the data stream 190. Accordingly, the browser may be receiving data of a supplemental file before it receives the HTML data entry that contains the reference to that supplemental file. That is not a problem. Particularly, when the browser reaches the reference to the supplemental file that it has already started

downloading and caching in its UOC, the UOC simply forwards the cached data to the browser for rendering.

B. The Krishnan Reference

Krishnan is the secondary reference in the rejections of claims 1-22 and has been cited for a very narrow proposition, namely, teaching "indicating an order". The Examiner has referred only to col. 3, line 45 to col. 4, line 9, which reads:

> In FIG. 1, a WEB browser application 101 is shown executing on a client computer system 102, which communicates with a server computer system 103 by sending and receiving HTTP packets (messages). The WEB browser application 101 requests WEB pages from other locations on the network to browse (display) what is available at these locations. This process is known as "navigating" to sites on the WEB network. In particular, when the WEB browser application 101 "navigates" to a new location, it requests a new page from the new location (e.g., server computer system 103) by sending an HTTP-request message 104 using any well-known underlying communications wire protocol. HTTP-request message 104 follows the specific layout discussed above, which includes a header 105 and a URI field 106, which specifies the target network location for the request. When the server computer system machine specified by URI 106 (e.g., the server computer system 103) receives the HTTP-request message, it decomposes the message packet and processes the request. When appropriate, the server computer system constructs a return message packet to send to the source location that originated the message (e.g., the client computer system 102) in the form of an HTTP-response message 107. In addition to the standard features of an HTTP message, such as the header 108, the HTTP-response message 107 contains the requested WEB page 109. When the HTTP-response message 107 reaches the client computer system 102, the WEB browser application 101 extracts the WEB page 109 from the message, and parses and interprets the HTML code in the page (executes the WEB page) in order to display the document on a display screen of the client computer system 102 in accordance with the HTML tags.

This portion of Krishnan appears in the Background section of the patent and describes nothing more nor less than the use of a conventional HTTP request/serve exchange in Internet Protocol. It describes in basic terms how a client machine requests a Web page and a server serves the Web page in response thereto. It does not, nor is it intended to, describe anything but the basic well-known request/response protocol between a client and server on the Internet for serving Web pages.

This passage contains no mention of supplemental files within a Web page, let alone any mechanism for indicating the order in which such supplemental file are to be served. This passage does not contain any discussion of the order of anything, in fact.

C. Discussion

1. The Examiner has not Presented a Prima Facie Obviousness Case

The Examiner asserted that Anders teaches all of the limitations of the claims except the "indicates an order" limitation (presumably referring to the claim limitation that the data defining the Web page contains data indicating the order in which the supplemental files are to be served). However, the Examiner asserts that "Krishnan teaches indicates an order (see col. 3, line 45 – col. 4, line 9)" and that it would have been obvious "to implement the teachings of Krishnan into the computer system of Anders to have indicates an order because it would have provided specific functions that comprehensible arrangement among the separate elements of the group".

The Examiner's analysis is flawed. Krishnan (1) does not teach that for which it has been cited because it does not disclose a Web page containing order data indicating the order in which supplemental files are to be disclosed and (2) even if it did teach such order data, the offered motivation for the proposed combination of features is improper.

Specifically, the assertion that "Krishnan teaches indicates an order" presumably is intended to mean that Krishnan teaches that there is some data in his Web page that "indicates an order in which said supplemental files are to be served, said order data cimporising data other than the order in which said supplemental files appear in said code defining said Web page". Otherwise, Krishnan would not be relevant. However, the cited portion of Krishnan, column 3, line 45 to column 4, line 9 contains no mention whatsoever of the order in which any supplemental files within a Web page are served. It does not mention supplemental files within a Web page at all. Rather, column 3, line 45-column 4, line 9 is in the background section of Krishnan and describes in very basic terms how a client machine requests a Web page and a server serves the Web page in response thereto. It does not, nor is it intended to, describe anything but the basic well-known request/response protocol between a client and server on the Internet for serving Web pages.

Thus, not only does Krishnan not contain any disclosure relating to the ordering of serving of supplemental files referenced in a Web page, it does not appear to contain any disclosure about (1) supplemental files in any context or (2) the order of anything.

Thus, Krishnan does not teach that for which it has been cited.

Even if Krishnan did teach a technique for ordering something, the rejection does not describe the proposed combination. Hence, Applicant cannot possibly more specifically address the issue of obviousness as Applicant does not know what the Examiner is asserting is obvious. More specifically, and using claim 1 as an example, the final Office Action asserted (erroneously) that Anders teaches "parsing the code comprising the requested page to detect data within the code which said supplemental files are to be served" and "Anders does not explicitly teach indicates an order. However, Krishnan teaches indicates and order. It would have been obvious ... to implement the teachings of Krishnan into the computer system of Anders to have indicates an order because it would have provided specific functions that comprehensible arrangement among the separate elements of a group".

This assertion, however, contains no description of a proposed combination. It completely lacks any discussion of what teaching of Krishnan is to be combined with what teaching of Anders.

As such, it is not a proper obviousness rejection. An Examiner must explain what the proposed combination is in order to present a prima facie case of obviousness. The Examiner has not done so in this case.

Even further, the motivation asserted by the Examiner, i.e., "because it would have provided specific functions that comprehensible arrangement among the separate elements of a group", essentially is meaningless. Specifically, initially, Applicant does

not understand the asserted motivation as quoted above as there appears to be clerical, grammatical, and/or typographical errors in it. Nevertheless at least at some level, it appears that the Examiner is asserting that the motivation has something to do with a desire to have things in an arrangement or order. However, this motivation does not make sense in the context of any of the prior art, including Anders, or the present invention. Specifically, both the conventional technique for serving a Web page and Anders already provide an order for serving the supplemental files. Thus, the motivation cannot be a generic desire to place things in order. They already are in order in both Anders and the more conventional prior art. Oddly, of all of the prior art being discussed, it is Krishnan that contains no teaching of a technique for setting an order for serving supplemental files. Yet this seems to be exactly what it has been cited as teaching.

Accordingly, the rejection must fail because (1) the present rejection is incomprehensible, and (2) the present rejection does not even set forth a proposed combination of the two prior art references.

2. <u>The Prior Art Does Not Teach The Claimed Invention In Any Event</u>

Returning to the Anders reference, Anders discloses an interesting technique for organizing the supplemental files that are served. However, it is an entirely different technique than the one of the present invention.

Thus, actually contrary to the Examiner's position, Anders does, in fact, disclose a technique for ordering the serving of supplemental files of the present invention. It is just a different technique than claimed in the present application.

Anders' server builds the data stream 190 using the Publisher software module 210 (see Figure 11). See column 10, line 1 to column 12, line 64. Specifically note the following passages:

> The stream configurator 212 parses through a page stored in a content storage 213 to identify references to objects and their locations within the page. Column 10, lines 8-10.

This clearly states that the stream configurator 212 receives a pre-existing, stored Web page and parses it.

> The stream configurator 212 also receives information about the objects, such as the sequence in which the objects comprising the page are to be displayed, and produces a stream configuration template 214. Column 10, lines 13-16.

> 'User supplied display sequence information. This provides information
> 'to the Interleavor for the order in which to display the objects.
> Write a record for the Input File to the template file. Column 10, lines 37-41.

These two sentences acknowledge that there is separate, user supplied information that indicates the order in which objects are to be displayed.

> At state 220, the publisher 210 prepares a stream header 180 to indicate that the data to follow is prepared using the data format of the present invention. The publisher 210 proceeds to state 222 where it generates a stream configuration template 214 including information regarding objects defined by a page, <u>their locations on the page and their sequence of display</u>. At state 224, the publisher 210 prepares data definition entries 192 (FIG. 9) for each object defined by the page. Note that the publisher 210 may generate the template 214 and prepare the data definition entries 192 (FIG. 9) concurrently. At state 226, the publisher 210 interleaves object data packets 200 (FIG. 10) with data definition entries 192 (FIG. 9) according to the sequence defined by the stream configuration template 214 so as to form a data stream 190. Lastly, the publisher 210 appends an end

of stream indicator 191 to indicate the end of the data stream 190. Column 12, lines 48-64.

'User supplied display sequence information. This provides information 'to the Interleavor for the order in which to display the objects. Column 11, lines 43-46.

Note that these passages disclose that the publisher 210 receives the objects in the page and receives a user supplied sequence of display for the objects.

Anders's Publisher 210 collects all of the objects on the page into one data stream so that the client machine can make only one request and then receive the whole page, including all of its supplemental files, rather than making a request for every supplemental file, which slows down the process significantly in communication systems in which the upstream channel is much slower than the downstream channel.

Thus, in Anders, there is nothing that resembles the sequence number attribute embedded within the tag referencing the supplemental file. The server does not parse the Web page code being sent to the client to detect the sequence numbers. There are no sequence numbers in Anders. Rather, Anders' server builds the data stream 190 with the Publisher software module 210using user supplied order information. The information dictating the order is not embedded within the main Web page itself, as claimed, but is separately supplied to the Publisher and the Publisher 210 dictates the order (by virtue of creating the data stream). Thus, while Anders' technology does permit the server to dictate the order in which supplemental files are delivered to the browser, it does so in a way that is entirely different than what is claimed in the present application.

Referring specifically to the language of claim 1, Anders does not disclose (1) "parsing the code comprising the requested page to detect data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page".

Thus, claim 1 patentably distinguishes over Anders by reciting the step of "parsing said code defining said Web page to detect order data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page".

Independent claim 9 also distinguishes over Anders. Claim 9 includes the limitation of "second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file". Hence, claim 9 recites similar features as found in claim 1 discussed above, but in language of differing scope. Therefore, claim 9 distinguishes over Anders for at least all of the reasons discussed above in connection with claims 1. However, claim 9 goes further and recites that the order data is an attribute of a tag. There is nothing in Anders remotely resembling this feature. The Examiner asserted that this is disclosed in col. 11, line 7-col. 12, line 44. However, it is quite clear from col. 11, lines 44-47 that the data identifying the display order does not come from HTML tags within the file. Col. 11, lines 44-47 state "User

supplied display sequence information. This provides information to the Interleavor for the order in which to display the objects".

Independent claim 12 also distinguishes over Anders by virtue of reciting "program code for parsing said code defining the Web page to detect said order data". Since, as previously discussed, the sequence information is not in the Web page in Anders, it obviously cannot retrieve that information by parsing the code of the Web page.

Anders also does not meet the limitation of claim 12 of "constructing a queue in a memory comprising a list of supplemental files in order". The order is given by the manner in which the data is sequenced in the packet. There is no list of the file order. This is an entirely different concept.

Accordingly, independent claim 12 patentably distinguishes over Anders.

Independent claim 19 includes the limitations "code for parsing said code defining a Web page to detect said order data", "code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order", and "code for serving said supplemental files to said requesting client machine in said order of said queue". In Anders, as previously mentioned, the display order is not found in the Web page and, therefore, these limitations are not met.

Furthermore, claim 19 specifically recites that the queue "compris[es] a list of said supplemental files in said order". Anders does not meet this limitation. In Anders, there is no list of the file order. The order is given by the manner in which the data itself

is interleaved in the packet. This is an entirely different concept.

Accordingly, claim 19 patentably distinguishes over Anders for many of the same reasons discussed above in connection with the other independent claims as well as these additional reasons.

3. The Dependent Claims

All of the dependent claims distinguish over the prior art for at least the reasons set forth above with respect to the independent claims from which they depend. However, the dependent claims also add even further patentably distinguishing recitations.

For instance, claim 7 depends from claims 1 and 2 and adds that the references to the supplemental files "comprise HTML tags, and said order data comprises attributes of said tags". This is similar to the limitation discussed above in connection with independent claim 9. As noted hereinabove, there is nothing in Anders remotely resembling this since the order is dictated by the Streaming Configurator.

Claim 8 depends from claim 7 and further adds that "said order data attributes are not recognizable by said client machine". This is directly contrary to Anders, in which the client machine must be modified in accordance with Anders' technology in order to recognize Anders' data stream.

Claim 11 depends from independent claim 9 and further distinguishes over Anders by further describing that the tag comprising the sequence number is an HTML

tag. Anders, which does not have a sequence number tag at all, obviously cannot teach such features.

Claim 17 depends from independent claim 12 and adds "said references to supplemental files comprise HTML tags" and that "said order data comprises attributes of said tags". As discussed above in connection with independent claim 9 (as well as dependent claim 7), these limitations are not found in Anders.

Claim 18 depends from claim 17 and further adds that "said order data attributes are not recognizable by said client machine". This is not found in Anders as discussed above in connection with claim 8.

Claims 21 and 22 depend from claim 19 and recite essentially the same subject matter as previously discussed in connection with dependent claims 7 and 8, respectively. Accordingly, they even further distinguish over the prior art for the same reasons given above in connection with dependent claims 7 and 8.

Respectfully submitted,

Dated: December 20, 2006

/Theodore Naccarella/
Theodore Naccarella, Reg. No. 33,023

Synnestvedt & Lechner
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189

S:\I\IBM\IBM Endicott\Patents\P24723 USA\PTO\Suuplemental_Appeal_Brief.doc

APPENDIX A: CLAIMS INVOLVED IN THIS APPEAL

1. A method of serving a Web page to a requesting client, said Web page comprising code defining said Web page and including a plurality of supplemental files, said method comprising the steps of:

parsing said code defining said Web page to detect order data within the code that indicates an order in which said supplemental files are to be served, said order data comprising data other than the order in which said supplemental files appear in said code defining said Web page;

constructing a queue indicating said order;

serving said code to said requesting client;

serving said supplemental files to said client in said order indicated in said queue.

2. The method of claim 1 further comprising the steps of:

receiving a request for said Web page; and

obtaining said code defining said Web page responsive to said request.

3. The method of claim 2 wherein said step of obtaining said Web page comprises retrieving said code defining said Web page from a memory.

4. The method of claim 2 wherein said step of obtaining said Web page comprises building said code defining said Web page responsive to said request.

5. The method of claim 2 further comprising the step of:

receiving and detecting requests from said client machine for said supplemental files; and

wherein said step of serving said supplemental files is performed after said receiving and detecting step.

6. The method of claim 2 wherein said step of serving said code defining said Web page is performed after said step of constructing said queue.

7. The method of claim 2 wherein said code defining said Web page comprises HTML code, said references to supplemental files comprise HTML tags, and said order data comprises attributes of said tags.

8. The method of claim 7 wherein said order data attributes are not recognizable by said client machine.

9. A computer readable storage medium containing executable code for controlling a computer for rendering a Web page, said code comprising:

first code at least partially defining said Web page, said code including a plurality of references to supplemental files containing content of said page; and

second code indicating an order in which said supplemental files are to be rendered, said second code associated with each of said references and comprising an attribute of a tag associated with said supplemental file.

11. The computer readable storage medium of claim 9 wherein said second code associated with each of said references comprises an attribute of an HTML tag for which another of said tag's attributes is said reference to a supplemental file.

12. A computer program product embodied on computer readable media readable by a computing device, said product for serving Web pages to a requesting client machine, wherein at least one of said Web pages contains a plurality of references to supplemental files comprising content of said Web page, said references including order data indicating an order in which said supplemental files are to be served relative to said other supplemental files contained in said page, said product comprising:

first computer readable program code for receiving requests for said Web pages;

second computer readable program code for obtaining code defining said requested Web pages responsive to said requests;

third computer readable program code for parsing said code defining said Web page to detect said order data;

fourth computer readable program code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order;

fifth computer readable program code for serving said code defining said Web page to said requesting client machine;

sixth computer readable program code for serving said supplemental files to said requesting client machine in said order of said queue.

13. The computer program product of claim 12 wherein said second computer readable program code comprises code for retrieving said code defining said Web page from a storage medium.

14. The computer program product of claim 12 wherein said second computer readable program code comprises code for building said code defining said Web page responsive to receipt of said request for said Web page.

15. The computer program product of claim 12 further comprising:

seventh computer readable program code for receiving and detecting requests from said client machine for said supplemental files and wherein said sixth computer

readable program code operates after said seventh computer readable program code detects said request for at least one of said supplemental files.

16. The computer program product of claim 12 wherein said fifth computer readable program code operates after said fourth computer readable program code constructs said queue.

17. The computer program product of claim 12 wherein:

said code defining said Web page comprises HTML code;

said references to supplemental files comprise HTML tags; and

said order data comprises attributes of said tags.

18. The computer program product of claim 17 wherein said order data attributes are not recognizable by said client machine.

19. A system for serving Web pages to a requesting client machine, at least one of said Web pages containing a plurality of references to supplemental files comprising content of said Web page, said page including order data indicating an order in which said supplemental files are to be served relative to said other supplemental files contained in said page, the system comprising:

a computer including memory, and a processor, the memory being accessible by the processor and storing computer-readable programming including,

first computer readable program code for receiving requests for said Web pages;

second computer readable program code for obtaining code defining said requested Web pages;

third computer readable program code for parsing said code defining said Web page to detect said order data;

fourth computer readable program code for constructing a queue in a memory, said queue comprising a list of said supplemental files in said order;

fifth computer readable program code for serving said code defining said Web page to said requesting client machine;

sixth computer readable program code for serving said supplemental files to said requesting client machine in said order of said queue.

20. The system of claim 19 wherein said fifth computer readable program code operates after said fourth computer readable program code constructs said queue.

21. The system of claim 19 wherein:

said code defining said Web page comprises HTML code;

said references to supplemental files comprises HTML tags; and

said order data comprises attributes of said tags.

22. The method of claim 21 wherein said order data attributes are not recognizable by said client machine.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None.